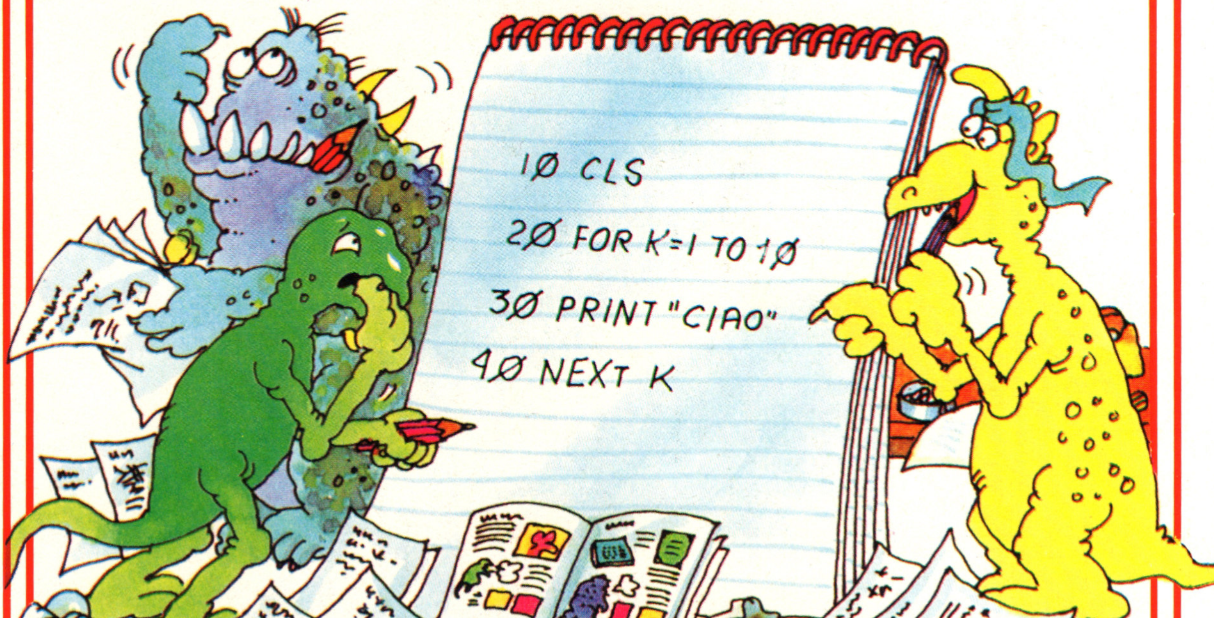
L. 2.500
Frs. 3,75

BEST-SELLER MONDIALE

LA GRANDE ENCICLOPEDIA DI INFORMATICA PER RAGAZZI

20

IN SOLI 30 FASCICOLI



GRUPPO EDITORIALE JACKSON

IN COLLABORAZIONE CON AMSTRAD

Spedizione in Abb. Postale Gruppo II/70

Direttore responsabile
Paolo Reina

Direttore di divisione:
Roberto Pancaldi

Autori:
Judy Tatchell,
Nick Cutler,

Lisa Watts,
Mike Wharton,

Tony Potter,
Ivor Guild,

Ian Graham,
Lynn Myring,

Helen Davies,
Mike Wharton,

Ian Graham,

Brian Reffin Smith,
Lisa Watts,

Bill Bennett,
Judy Tatchell,

Jenny Tyler,

Lee Howarth,
Judy Tatchell,

Gaby Waters,
Graham Round,

Nick Cutler,
Gaby Waters,

Brian Reffin Smith,

Judy Tatchell,
Lee Howarth,

Cherry Evans,
Lee Howarth

Revisione e adattamento:
Martino Sangiorgio

Coordinamento editoriale:
Renata Rossi

Progetto grafico:
Sergio Mazzali

Distribuzione:
SODIP - Milano

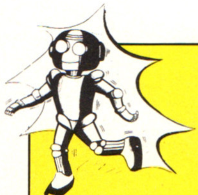
Stampa:
Vela - WEB - Vigano di Gaggiano (MI)

Direzione e Redazione:
Via Rosellini, 12 - Milano (20124) - Tel. 02/6880951 (5 linee)

Autorizzazione alla pubblicazione: Tribunale di Milano n° 226 del 28/3/89.
Spedizione in abbonamento postale Gruppo II/70
(autorizzazione della Direzione Provinciale delle PPTT di Milano)
Prezzo del fascicolo L. 2.500
I numeri arretrati saranno disponibili per 1 anno dal completamento dell'opera e potranno essere richiesti direttamente all'Editore a L. 3.000 (sovrapprezzo di L. 10.000 per spese d'imballo e spedizione).
I versamenti vanno indirizzati a:

Gruppo Editoriale Jackson S.p.A.
Via Rosellini, 12 - 20124 Milano

mediante emissione di assegno bancario oppure utilizzando il C.C. Postale N. 11666203.
Non vengono effettuate spedizioni in contrassegno.



NEL PROSSIMO NUMERO:

- CHE COS'È IL LINGUAGGIO MACCHINA
- PROGRAMMA DI CACCIA AL TESORO
- CHE COS'È UN GIOCO FANTASY
- LAVAGNA LUMINOSA COMPUTERIZZATA
- INPUT/OUTPUT VOCALE

La pulce ballerina

In questo gioco voi controllate una pulce ballerina che si muove per lo schermo divorando le erbacce che hanno infestato il vostro giardino. Così come sta il programma usa i normali caratteri della tastiera per rappresentare una pulce (X) ed un'erbaccia (#). Potete creare le vostre "shape" per la pulce e per le erbacce se il vostro computer ve lo consente. Potete trovare di più su come fare le vostre "shape" guardando nel vostro manuale sotto caratteri definiti dall'utente.

Come giocare alla Pulce ballerina

Quando fate girare questo programma il vostro giardino viene disegnato sullo schermo, e si riempie gradualmente di sempre più numerose erbacce. Premete la barra dello spazio per iniziare il gioco: la vostra prima pulce appare e la potete muovere in alto, in basso, a sinistra e a destra premendo i tasti "A", "Z", "," e ".". La pulce salta anche oltre i muri del giardino. Quando colpisce un'erbaccia questa scompare: avete un punteggio per ogni erbaccia mangiata. Se una pulce mangia tutte le erbacce prima che scada il tempo, ottenete anche un punteggio premio addizionale. Se no, perdete quella pulce ed un'altra pulce appare sullo schermo. In ciascun gioco avete a disposizione tre pulci; il tempo a disposizione per mangiare le erbacce diminuisce con ciascuna pulce.

C'è un buco invisibile nel giardino: se la pulce cade dentro il buco la perdete. Il buco è in un luogo diverso ogni volta.

Dopo che avete perso tutte le tre pulci il computer vi domanda se volete giocare ancora e voi premete S o N.

```
•5 WIDTH 40:RANDOMIZE
```

WIDTH 40 determina la risoluzione in modo testo (massimo 40 caratteri per linea e 25 linee)

```
•7 COLOR INT(RND(1)*3),INT(RND(1)*2)

•10 LET WI%=39
•20 LET HI%=23
```

I valori di WI% e HI% dipendono dalla risoluzione adottata. Corrispondono al numero massimo meno 1 di caratteri visualizzabili per ogni riga e per ogni colonna (per i valori di riga, cioè HI% è meglio togliere 2 per problemi di scroll del video).

```
 30 LET V%=1
 40 LET W%=0
 50 LET G%=15:LET Z=1
 60 LET TS%=0
 70 LET NB%=3
 80 DIM X%(G%),Y%(G%)
 90 GOSUB 830
 100 GOSUB 550
 110 GOSUB 800

•120 BEEP:GOTO 270
```

BEEP fa suonare l'altoparlante per 1/4 di secondo a 830 Hz. Può essere sostituita da una SOUND.

```
 130 LET Q$=INKEY$

 140 IF Q$="A" THEN LET W%=-1:LET V%=0
 150 IF Q$="Z" THEN LET W%=1:LET V%=0
 160 IF Q$="," THEN LET V%=-1:LET W%=0
 170 IF Q$="." THEN LET V%=1:LET W%=0
```

Tasti di direzione della pulce.

```
 180 GOSUB 440
 190 LOCATE Y%,X%:PRINT " "
 200 LET X%=X%+V%
 210 LET Y%=Y%+W%

 220 IF X%<2 THEN LET X%=2:LET V%=-V%
 230 IF Y%<4 THEN LET Y%=4:LET W%=-W%
 240 IF X%>WI%-1 THEN LET X%=WI%-1:LET V%=-V%
 250 IF Y%>HI%-1 THEN LET Y%=HI%-1:LET W%=-W%
```

Controllano se la pulce ha raggiunto un muro, nel qual caso la fanno tornare indietro.

```
 260 LOCATE Y%,X%:PRINT CHR$(D%)
 270 FOR I%=1 TO G%
 280 IF X%=X%(I%) AND Y%=Y%(I%) THEN GOSUB 490
 290 NEXT I%
```

```
●300 IF X%=P% AND Y%=Q% THEN BEEP:LET T%=1
310 LOCATE Y%,X%:PRINT CHR$(88)
320 LET T%=T%-1
330 LOCATE 2,1:PRINT "PUNTEGGIO ";TS%;" TEMPO ";T%;" "
340 IF T%>0 THEN GOTO 130
350 IF B%>0 THEN LOCATE 5,5:PRINT "BONUS ";B%
360 FOR C%=1 TO 5000:NEXT C%
370 LET TS%=TS%+B%
380 IF B%=0 THEN LET NB%=NB%-1
390 IF NB%>0 THEN GOTO 100
400 LOCATE 5,5:PRINT "GIOCHI ANCORA? (S/N)"
410 INPUT Q$
420 IF Q$="S" THEN RUN
430 STOP
440 IF W%=1 THEN LET D%=86
450 IF W%=-1 THEN LET D%=94
460 IF V%=-1 THEN LET D%=60
470 IF V%=1 THEN LET D%=62
480 RETURN
490 LET S%=S%+1
500 LET TS%=TS%+5
510 LET X%(I%)=1
520 LET Y%(I%)=1
530 IF S%=G% THEN LET B%=T%/3:LET T%=1
540 RETURN
550 CLS
570 LOCATE 1,1:PRINT "LA PULCE BALLERINA : PULCI ";NB%
580 LET S%=0
590 LET B%=0
600 LET Z=Z+.3
610 LET T%=1000/Z
620 FOR I%=1 TO G%
640 GOSUB 800
650 LET X%(I%)=X%:LET Y%(I%)=Y%
660 LOCATE Y%(I%),X%(I%):PRINT CHR$(35)
670 NEXT I%
680 GOSUB 800
690 LET P%=X%:LET Q%=Y%
710 FOR I%=1 TO WI%
720 LOCATE 3,I%:PRINT "*"
730 LOCATE HI%,I%:PRINT "*"
740 NEXT I%
750 FOR I%=3 TO HI%-1
760 LOCATE I%,1:PRINT "*"
770 LOCATE I%,WI%:PRINT "*"
780 NEXT I%
790 RETURN
800 LET X%=RND(1)*(WI%-3)+2
810 LET Y%=RND(1)*(HI%-5)+4
820 RETURN
830 GOSUB 550
840 GOSUB 620
860 LOCATE 2,1:PRINT "PREMERE 'SPAZIO' PER PARTIRE"
870 LET Q$=INKEY$
880 IF Q$<>" " THEN GOTO 840
890 RETURN
```

Riga 300: BEEP fa suonare l'altoparlante per 1/4 di secondo a 830 Hz. Può essere sostituita da una SOUND.

Righe 440–470: Visualizzano un diverso simbolo in funzione della direzione assunta dalla pulce.

Riga 530: Se S%=G% vuol dire che la pulce ha estirpato tutte le erbacce. In tal caso viene calcolato un bonus (B%) uguale a un terzo del tempo rimasto.

Righe 710–780: Disegnano il terreno di gioco.

Righe 840–890: Routine di attesa per permettere al giocatore di prepararsi. Quando è pronto preme la barra spazio e il gioco inizia.

Variazioni sui giochi

Ecco alcune idee su come giocare con i giochi elettronici in modo diverso, per renderli piú difficili o per giocarvi con piú di una persona. La maggior parte di queste idee è applicabile anche ai giochi da bar, sempre che non ti dispiaccia spendere un po' di soldi in piú.

"Sopravvivenza"



Scopo di questo gioco è sopravvivere, non fare punti. Se giochi contro altre persone, vince chi sopravvive piú a lungo. Se qualcuno diventa troppo bravo, fallo giocare a mani incrociate.

Mani incrociate



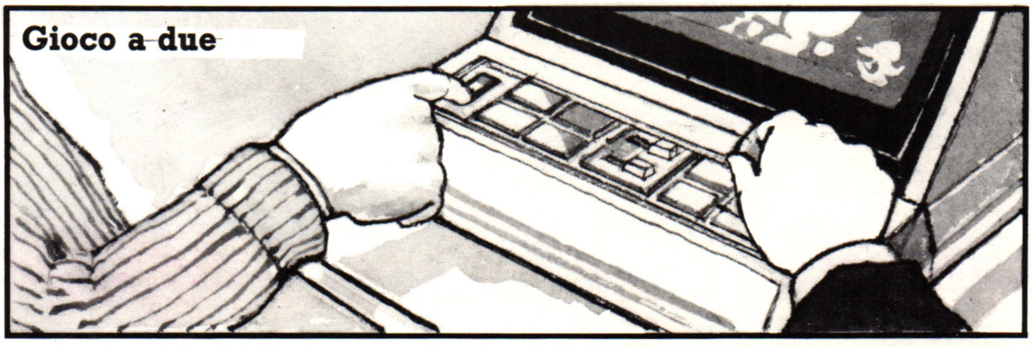
Per rendere un gioco piú difficile, prova a incrociare le mani, in modo che quella con cui abitualmente spari sia sull'altro comando. E' piuttosto complicato, quindi comincia a giocare dal livello piú semplice.

"Suicidio"



E' l'opposto del gioco di sopravvivenza: scopo è di finire la partita il piú rapidamente possibile con il minor numero di punti.

Gioco a due

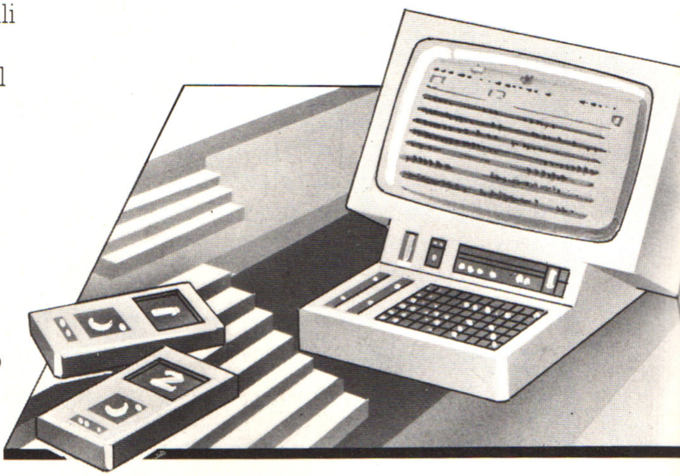


Con due persone che giocano lo stesso gioco, una può controllare il pulsante di sparo e l'altra la base laser o qualsiasi altra cosa il gioco preveda.

Alternativamente, una persona può controllare tutti i comandi e l'altra guardare lo schermo e dire cosa va fatto. Per questo gioco occorrono riflessi molto rapidi.

Giochi del futuro

Probabilmente fra dieci anni gli attuali giochi su computer sembreranno molto semplici e primitivi; i giochi del futuro saranno piú veloci e interessanti, con immagini sullo schermo incredibilmente realistiche. Questi giochi conterranno computer cento volte piú potenti e veloci di quelli odierni e in grado di immagazzinare milioni di elementi informativi. Quei computer saranno ottenuti inserendo in un chip di silicio piú circuiti sempre piú complicati. Ecco alcune prevedibili caratteristiche dei giochi su computer di un futuro non lontano.



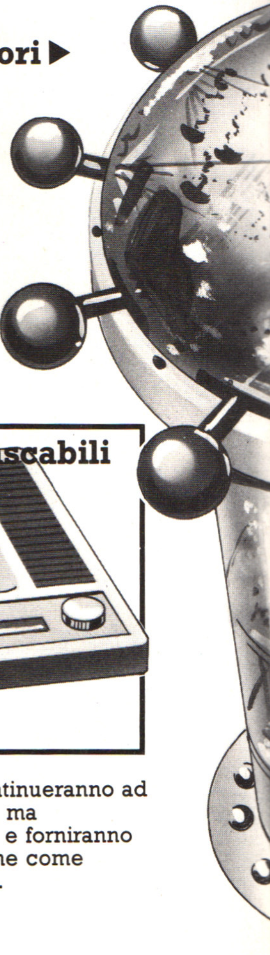
Potenti computer potranno creare giochi di avventura enormemente piú complessi di quelli attuali. Per aiutare il giocatore a scegliere e registrare le mosse ci saranno probabilmente una tastiera e dei contatori.



Un videogioco con un'ampia memoria sarà in grado di ricostruire immagini dettagliate, per esempio della battaglia di Waterloo o di una battaglia spaziale, e i giocatori potranno controllare molti piú aspetti del quadro che non oggi.

Gioco per piú giocatori ▶

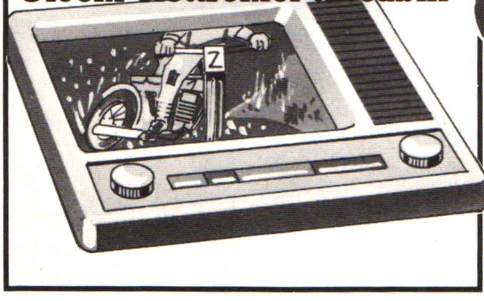
Attualmente la maggior parte dei giochi su computer sono solo per un giocatore o due. I computer piú potenti saranno invece in grado di eseguire istruzioni di piú persone che potranno giocare gli uni contro gli altri o contro il computer.





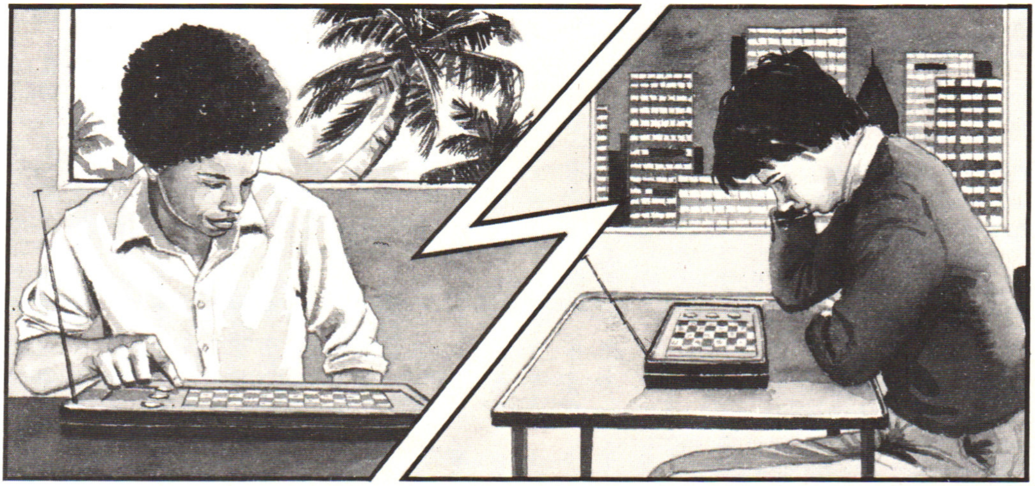
Nei videogiochi sportivi si potrà probabilmente controllare ciascun membro della squadra individualmente. Questi giochi avranno anche voci elettroniche sintetizzate e l'arbitro dirà quando un giocatore è fuori gioco o che punizione è stata data.

Giochi elettronici tascabili



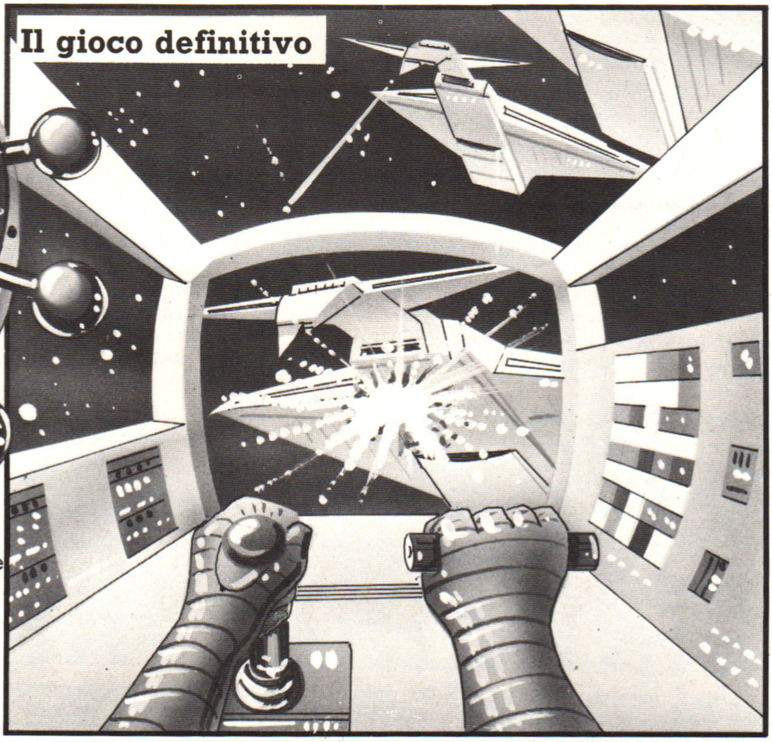
I giochi elettronici tascabili continueranno ad avere schermi a cristalli liquidi, ma probabilmente saranno a colori e forniranno immagini dettagliate e realistiche come quelle dei programmi televisivi.

Giochi a distanza



Nel duemila potrai sfidare qualcuno che abita a centinaia di chilometri di distanza. I giochi conterranno radiotrasmittenti e riceventi miniaturizzate che trasmetteranno le tue mosse e riceveranno quelle del tuo avversario quasi immediatamente. Le mosse del tuo avversario verranno eseguite automaticamente sullo schermo a cristalli liquidi.

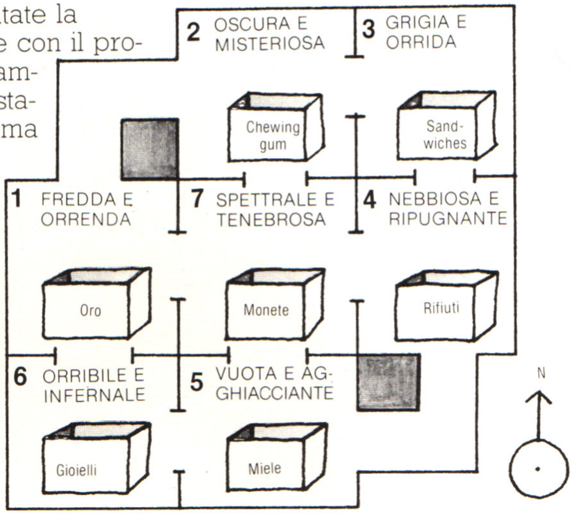




Il gioco definitivo sarà una simulazione superrealistica che si svolgerà tutto intorno al giocatore in una cella speciale per i giochi. Il gioco, forse un'invasione spaziale o uno di avventura, avrà effetti tridimensionali, luci laser e suono quadrifonico.

Come scrivere un programma di caccia al tesoro

Nelle prossime pagine troverete dettagliate istruzioni per scrivere un gioco di caccia al tesoro che consiste nel muoversi attraverso un labirinto di sette stanze per raccogliere un tesoro. Il programma è un po' complicato per cui seguite le istruzioni attentamente e provate le varie parti intanto che procedete. Se vi bloccate in qualche parte del programma, consultate la risposta a quel problema, quindi continuate con il programma che rimane. Per scrivere il programma avete bisogno di buone idee per impostare il gioco, perciò leggete questa parte prima di iniziare.



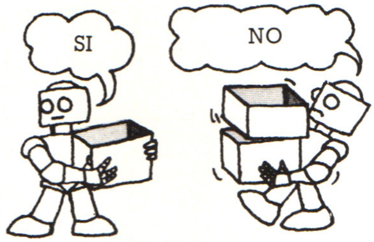
Come giocare

La mappa sulla destra mostra il labirinto di sette stanze. Ciascuna ha una descrizione diversa e contiene una scatola con un tesoro. L'informazione mostrata in questa mappa è memorizzata nella memoria del computer, ma la mappa non appare sullo schermo. Questo significa che dovrete indovinare la posizione di ogni stanza di questa mappa mentre giocate. Iniziando da una stanza scelta a caso, lo scopo del gioco è di raccogliere tutto il tesoro in una stanza in un limitato numero di mosse.

```
QUESTE SONO LE PAROLE
CHE IL COMPUTER ACCETTA
N,E,S,O : MUOVE VERSO NORD/EST/SUD/OVEST

PRENDI  : RACCOGLIE IL TESORO
LASCIA  : DEPONE IL TESORO
POSTO   : STAMPA LA POSIZIONE
          ATTUALE DEL TESORO
AIUTO   : VI DICE COME GIOCARE
```

Lo schermo sopra descrive i comandi che potete usare per giocare ed il loro significato. È possibile inserire un solo comando per ogni mossa ed allora non potete permettervi di sbagliare.



Per rendere il vostro compito piú difficile, vi è consentito trasportare una scatola di tesoro alla volta.

Esempio del programma

1

```
SIETE NELLA STANZA  4
E' NEBBIOSA E RIPUGNANTE
CONTIENE :      RIFIUTI

COSA VOLETE FARE ?
? PRENDI
O.K. STATE TRASPORTANDO RIFIUTI
```

2

```
SIETE ANCORA NELLA STANZA  4
COSA VOLETE FARE ?
? O
```

3

```
SIETE NELLA STANZA  7
E' SPETTRALE E TENEBROSA
CONTIENE :      MONETE

COSA VOLETE FARE ?
? LASCIA
RIFIUTI NELLA STANZA  7
```

4

```
SIETE ANCORA NELLA STANZA  7
COSA VOLETE FARE ?
? S

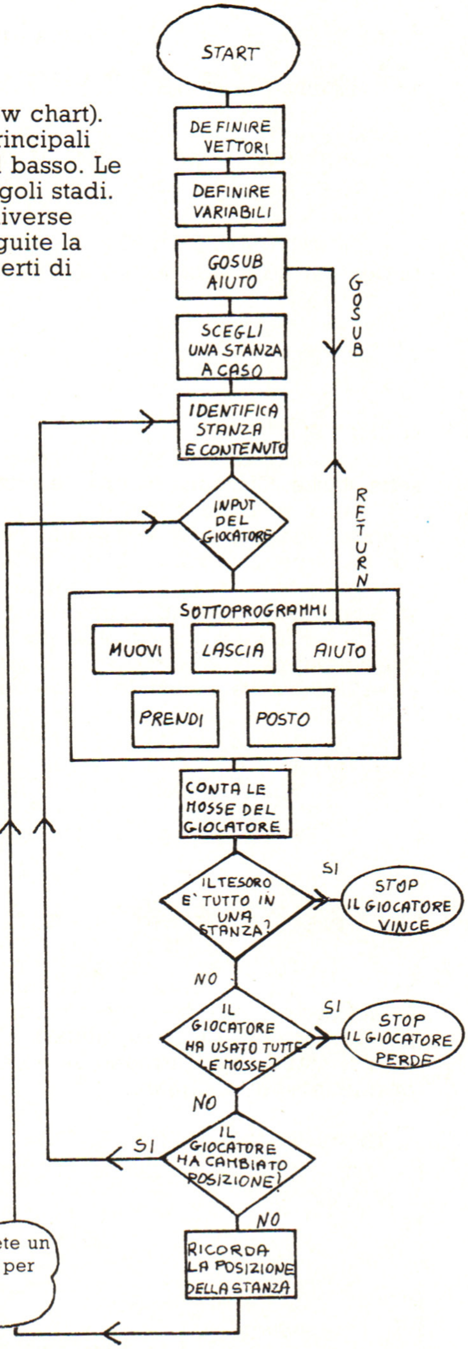
SIETE NELLA STANZA  5
```

Per scrivere il programma

La figura a destra rappresenta una carta di flusso (flow chart). Essa illustra la struttura del programma i cui punti principali sono elencati in ordine procedendo dall'alto verso il basso. Le istruzioni che seguono vi guideranno attraverso i singoli stadi. Non preoccupatevi se vi sarà detto di saltare nelle diverse parti del programma senza rispettare l'ordine. Se seguite la numerazione delle righe consigliata, potete essere certi di assemblare il programma nella giusta sequenza.

1 Definire i vettori e i dati (righe 100-250)

Dovete mettere tutte le informazioni contenute nella mappa della pagina accanto nella memoria del computer. Per fare questo, servitevi di un certo numero di vettori.



Vettori N, E, S, O

Avete prima bisogno di quattro vettori che chiameremo N, E, S e O. I dati di questi vettori dicono al computer quale stanza è al Nord, ad Est, al Sud e ad Ovest di ogni altra stanza del labirinto. Dimensionate questi vettori nella prima riga (100) ; ognuno di essi ha sette elementi.

	N	E	S	O
1	2	7	6	0
2	0	3	7	1
3				
4				
5				
6				
7				

0 significa che non ci sono stanze in quella direzione. Per esempio, non c'è una stanza ad Ovest della 1 o a Nord della 2.



Create una tabella come questa per completare i dati. I numeri della prima colonna sono gli argomenti dei vettori e rappresentano i numeri delle stanze. Per ogni riga della tabella controllate la mappa delle stanze ed inserite il numero della stanza che sta a Nord, ad Est, a Sud e ad Ovest delle stanze elencate nella prima colonna. Le prime due righe sono state già completate.

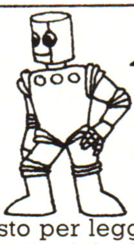
Aggiungete un comando per pulire lo schermo.



Per leggere i dati

```
110 FOR K=1 TO 7
120 READ N(K),E(K),S(K),U(K)
130 NEXT K

2000 DATA 2,7,6,0
```



Fate attenzione che ad ogni READ corrisponda effettivamente una DATA, altrimenti potreste avere risultati imprevedibili o degli errori di programma.

Adesso avete bisogno di un ciclo come questo per leggere i dati e immetterli nei vettori. Usate la tabella per mettere i dati in sette righe DATA. La prima di esse è mostrata sopra.

Il vettore D$

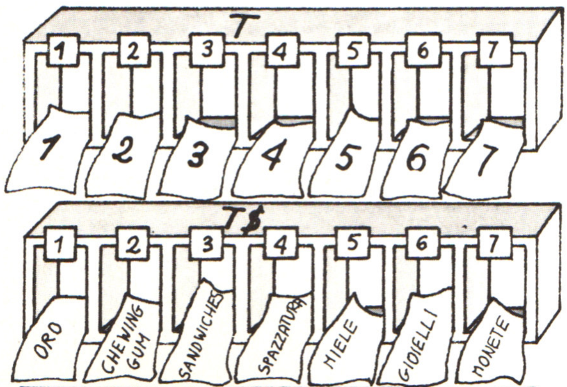
```
2100 DATA "FREDDA E ORRENDA" "OSCURA E M
ISTERIOSA"
```

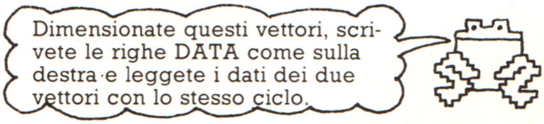


Per memorizzare le descrizioni delle stanze avete bisogno di un vettore che potrete chiamare D$. Dimensionatelo all'inizio del programma e scrivete i dati, come mostrato sopra, servendovi delle descrizioni della mappa.

I vettori T$ e T

Avete adesso bisogno di due altri vettori; T$ per ricordare i nomi del tesoro e T che vi dice dove si trova ciascun tesoro. Il numero immagazzinato in ogni casella di T rappresenta il numero della stanza dove è archiviato il tesoro che si trova nella casella corrispondente di T$. Per esempio all'inizio del gioco, T(2) vale 2 e questa è la stanza dove è conservato il tesoro T$(2) (il chewing-gum). Durante il gioco i numeri di T cambiano in quanto le scatole del tesoro vengono spostate.






```
2200 DATA ORO,1,"CHEWING GUM",2
2210 DATA SANDWICHES,3,RIFIUTI,4
```

Le variabili

M	C	F	WXY
Numero di mosse fatte dal giocatore.	Dice se il giocatore sta portando qualcosa.	Una variabile "bandiera". Capirete il funzionamento dopo.	Variabili che memorizzano dei dati temporanei.

In questi riquadri trovate i nomi delle variabili che vi serviranno successivamente nel programma. A partire dalla riga 300, date un valore iniziale O a queste variabili.

2 Sottoprogramma 'Aiuto' (righe 1000-1120)

```
CI SONO 7 STANZE NEL LABIRINTO
E IN OGNUNA DI ESSE C'E' UNA
SCATOLA CON UN TESORO. DOVETE
PORTARE TUTTO IL TESORO IN UNA
SOLA STANZA.
```

Ora spiegate le regole del gioco con un sottoprogramma. Mettete un comando GOSUB alla riga 300 e iniziate il sottoprogramma alla riga 1000. Scrivete le istruzioni mostrate sullo schermo a sinistra e le parole d'ordine che il computer accetta

3 Scegliere una stanza a caso (riga 350)

Il numero della stanza dove si trova il giocatore è memorizzato dalla variabile R. Per stabilire da quale stanza iniziare, fate scegliere al computer un numero a caso compreso tra 1 e 7 e memorizzatelo in R alla riga 350.

4 Identificare la stanza ed il contenuto (righe 400-470)

```
SIETE NELLA STANZA  7
E' SPETTRALE E
TENEBROSA
CONTIENE :     MONETE
```

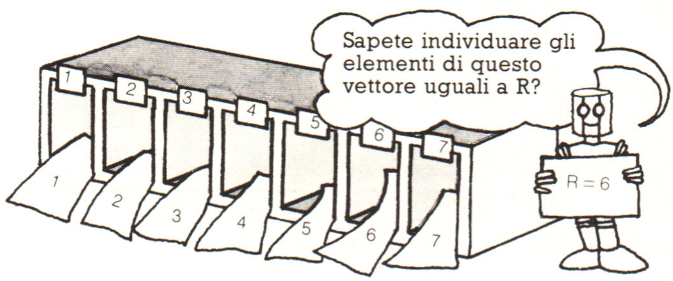
```
SIETE NELLA STANZA  4
E' NEBBIOSA E
RIPUGNANTE
CONTIENE :   RIFIUTI
```

```
SIETE NELLA STANZA  1
E' FREDDA E ORRENDA
CONTIENE :     ORO
```

Informate innanzitutto il giocatore in quale stanza si trova. Descrivete poi la stanza usando R come argomento in D$. Gli schermi in alto illustrano come dovrebbe apparire il vostro schermo.

Identificare il contenuto di una stanza

```
430 FOR K=1 TO 7
440 IF T(K)=R THEN PRINT TAB(15);T$(K):
LET F=1
450 NEXT K
460 IF F=0 THEN PRINT TAB(15);"NIENTE"
470 LET F=0
```



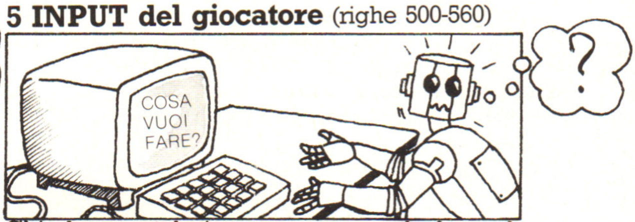
Per sapere il contenuto della stanza R cercate nel vettore T che memorizza il numero della stanza dove c'é ciascuna scatola del tesoro. Per fare questo usate un ciclo come sopra. La riga IF/THEN controlla se un numero qualunque di T coincide con il numero di stanza R; se sí il computer stampa il nome della scatola del tesoro e pone la variabile F = 1.

La variabile "bandiera"



La variabile F dice al computer se c'è qualche tesoro nella stanza dopo aver eseguito il ciclo. Essa funziona quasi come una bandiera. Quando la riga IF/THEN è vera la bandiera è su (cioè F è 1); in caso contrario F resta uguale a zero. Ricordatevi di rimetterla sempre a zero (riga 470) non appena la userete di nuovo.

5 INPUT del giocatore (righe 500-560)



Chiedete ora al giocatore cosa vuole fare ed inserite poi una riga INPUT che permette al giocatore di battere un comando (come PRENDI, LASCIA ecc.) come una variabile A$. Controllate l'INPUT del giocatore scrivendo cinque righe GOSUB che indirizzeranno il computer verso differenti sottoprogrammi per ogni comando. Mettete i numeri di riga quando avrete deciso in quale parte del programma vanno i sottoprogrammi.

6 Il sottoprogramma "Muovi" (righe 1200-1260)

Provate a scrivere il sottoprogramma usando questo flow-chart come guida.

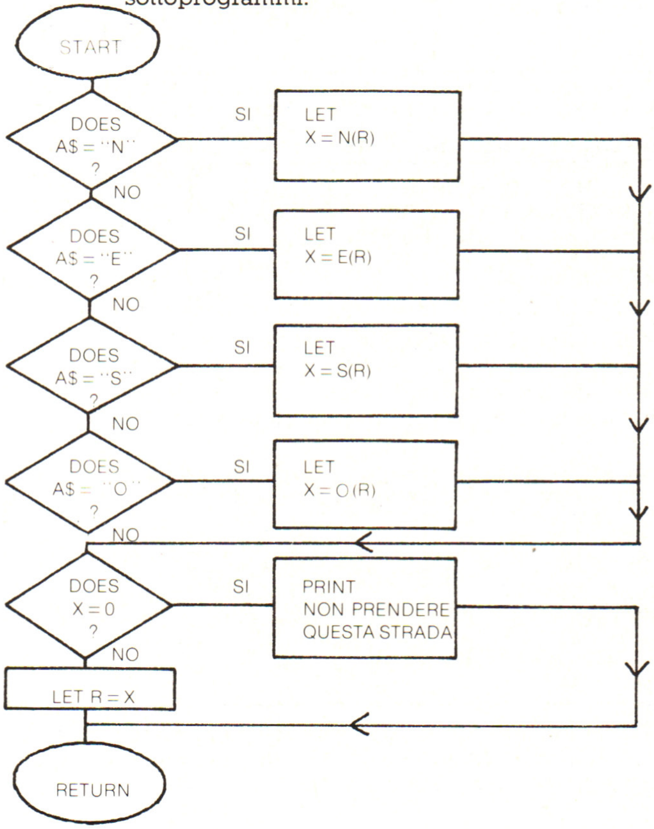
1. Per muovere, il giocatore batte N, E, S o O; controllate quindi quale di questi è stato battuto mediante quattro righe IF/THEN.

2. Per trovare il nuovo numero di stanza, usate R come argomento del vettore corrispondente (N,E,S o O) e memorizzatelo in una variabile temporanea X.

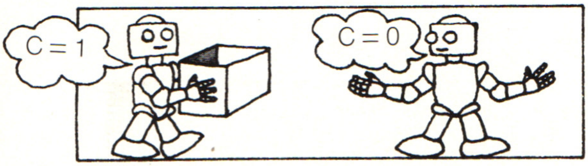
3. Dovete poi controllare il valore di X per essere certi che il giocatore può muoversi in quella direzione. Se X è zero, non ci sono stanze in quella direzione; dovrete quindi stampare un messaggio ed usare GOTO per mandare il computer all'istruzione RETURN in fondo al sottoprogramma.

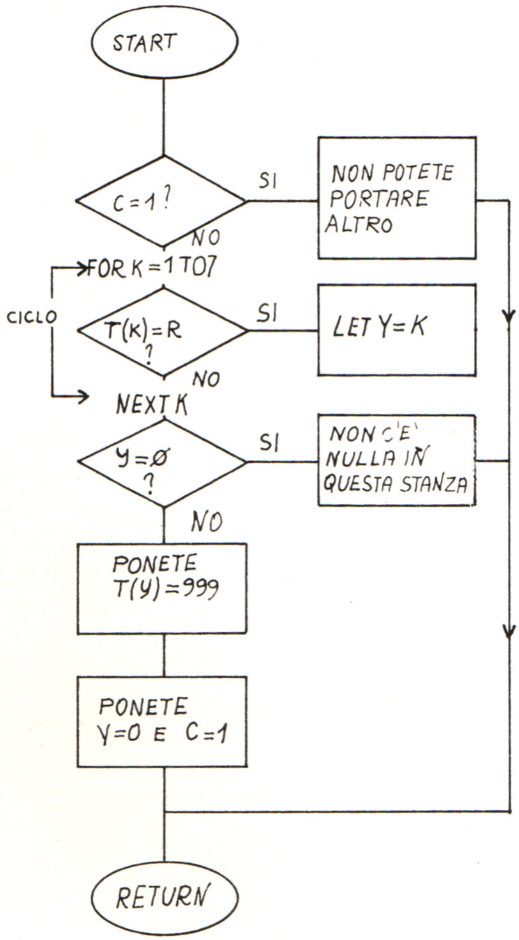
4. Se X è diverso da zero, dovrete mettere il nuovo numero di stanza in R, ponendo R uguale a X.

5. Ponete RETURN alla fine del sottoprogramma.

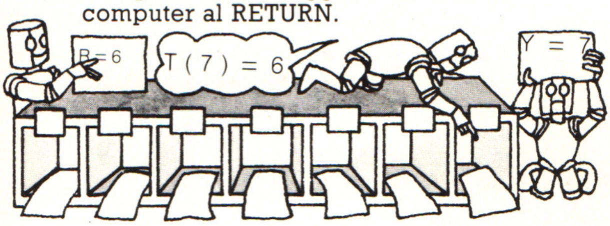


7 Sottoprogramma 'Prendi'
(righe 1300-1380)





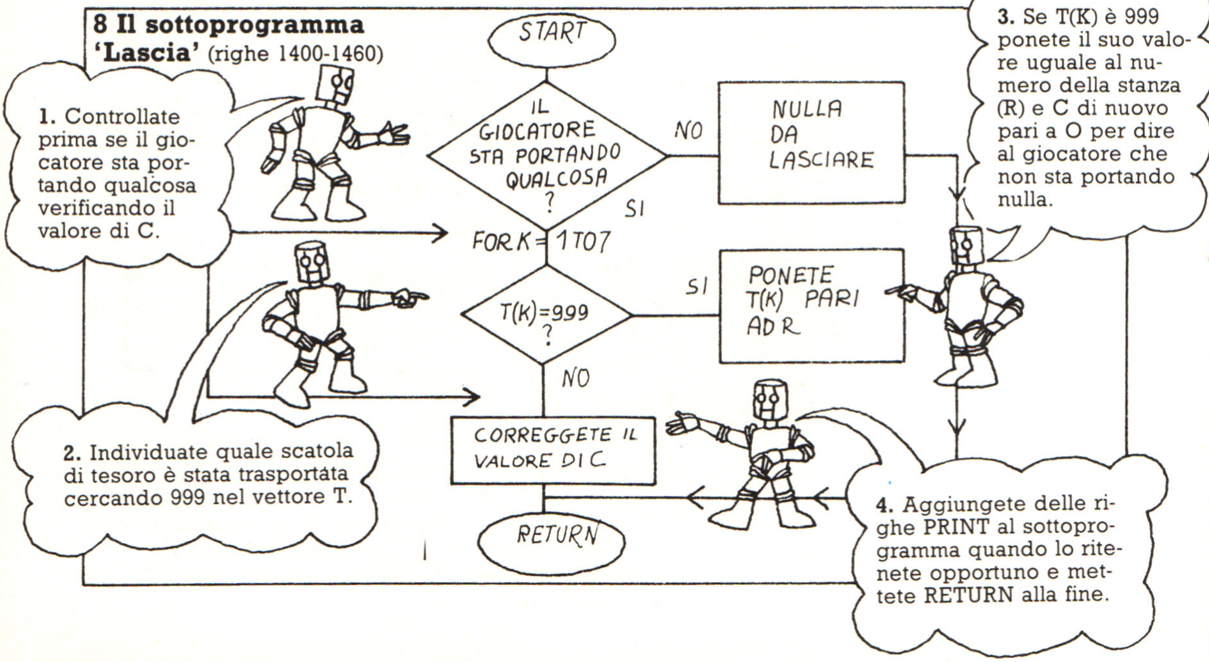
1. Il giocatore può trasportare solo una scatola di tesoro per volta, controllate quindi se il giocatore sta portando qualcosa. Per fare questo, verificate il valore di C che è pari a 1 quando il giocatore prende il tesoro. Se C è 1, stampate un messaggio e indirizzate il computer al RETURN.



2. Per controllare se c'è un tesoro nella stanza, scrivete un ciclo usando IF/THEN per cercare nel vettore T. Se un numero di T è uguale al numero di stanza (R), dovreste memorizzare l'argomento di T (vedi K) in una variabile temporanea detta Y. Quando il computer gira all'interno del ciclo, può trovare diversi numeri uguali a R. In questo caso, Y conterrà l'argomento dell'ultimo numero di T, quando il computer avrà completato il ciclo.

3. Se Y è O, non c'è alcun tesoro nella stanza e dovrete indirizzare il computer al RETURN; altrimenti l'argomento Y dice al computer quale scatola di tesoro prendere. Ponete il valore di T(Y) pari a 999, un numero che dimostra che esso è stato trasportato. Ponete C = 1 e di nuovo Y = O; mettete RETURN in coda al sottoprogramma.

8 Il sottoprogramma 'Lascia' (righe 1400-1460)



1. Controllate prima se il giocatore sta portando qualcosa verificando il valore di C.

2. Individuate quale scatola di tesoro è stata trasportata cercando 999 nel vettore T.

3. Se T(K) è 999 ponete il suo valore uguale al numero della stanza (R) e C di nuovo pari a O per dire al giocatore che non sta portando nulla.

4. Aggiungete delle righe PRINT al sottoprogramma quando lo ritenete opportuno e mettete RETURN alla fine.

9 Il sottoprogramma "Posto" (righe 1500-1590)

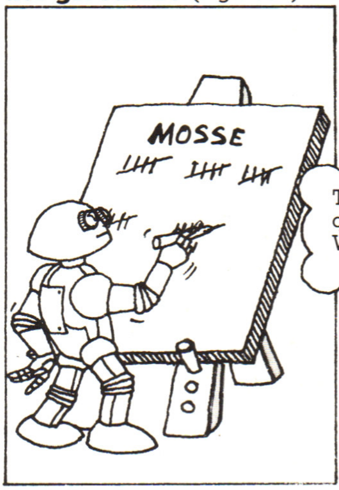
Questa routine fornisce al giocatore la posizione attuale di ogni scatola del tesoro. Cercate di scrivere il programma servendovi di quanto scritto sui due schermi in basso. Progettate ciò su un diagramma di flusso.

```
STATE TRASPORTANDO : ORO
CONTENUTI DELLE STANZE :
 2  : CHEWING GUM
 3  : SANDWICHES
 4  : RIFIUTI
 5  : MIELE
 6  : GIOIELLI
 7  : MONETE
```

```
STATE TRASPORTANDO : NIENTE
CONTENUTI DELLE STANZE :
 7  : ORO
 7  : CHEWING GUM
 3  : SANDWICHES
 7  : RIFIUTI
 7  : MIELE
 7  : GIOIELLI
 7  : MONETE
```

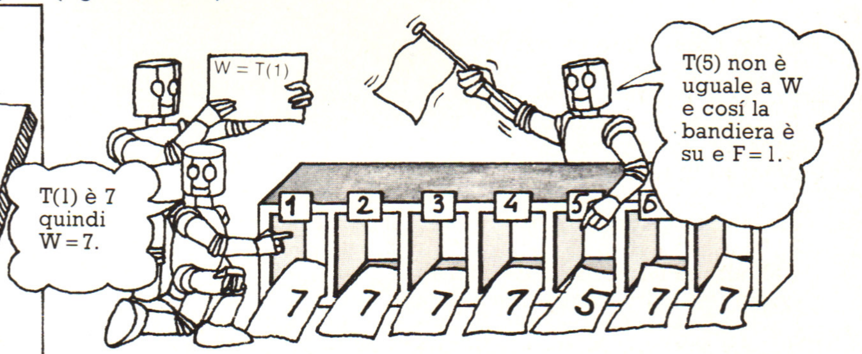
Controllate prima il valore di C per vedere se il giocatore sta portando qualcosa. Se sí, indirizzate il computer ad un ciclo per trovare nel vettore T l'elemento che uguaglia 999, quindi stampate il nome della scatola del tesoro che sta per essere trasportata selezionando l'elemento del vettore T$ con lo stesso argomento. Per stampare i contenuti delle stanze occorre un altro ciclo che ne scriva le attuali posizioni ad eccezione della scatola che viene trasportata (se ce n'è una).

10 Per contare le mosse del giocatore (riga 600)



Con la riga 600 aggiungete 1 alla variabile M per contare le mosse che fa il giocatore.

11 Il tesoro è tutto nella stessa stanza? (righe 610-690)



Il giocatore vince se riesce a portare tutto il tesoro nella stessa stanza. Potete verificare ciò controllando i numeri memorizzati nel vettore T poiché essi saranno tutti uguali quando il tesoro sarà nella stessa stanza. Per fare questo, prendete un elemento di T, per esempio T(1), e immagazzinatelo in una variabile temporanea detta W. Confrontatela poi con tutti gli altri numeri memorizzati in T, uno alla volta all'interno di un ciclo. Usate una variabile 'bandiera' per indicare quando essi sono tutti uguali o no.

12 Il giocatore ha usato tutte le mosse a disposizione? (righe 700-750)



Stabilite quante mosse sono concesse al giocatore. Se M è maggiore di questo numero il giocatore perde la partita. Quando si verifica ciò, stampate un messaggio e fermate il programma.

13 Il giocatore ha cambiato posto? (righe 800-850)

```
SIETE ANCORA NELLA STANZA  7
COSA VOLETE FARE ?
?
```

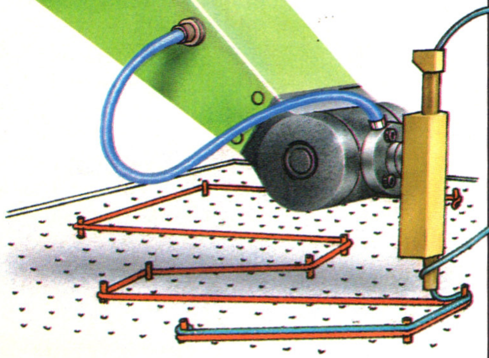
Se il giocatore ha battuto N,E,S o O in A$ nella riga INPUT, portate indietro il giocatore alla riga 400 per identificare la nuova stanza ed il suo contenuto. Altrimenti, ricordategli la posizione e indirizzate il computer alla riga INPUT (500).

Cosa un robot può o non può fare

I robot possono fare molte cose, specialmente in una fabbrica, dove vengono sottoposti a un'accurata manutenzione e sono predisposti per lavorare accanto ad altri macchinari automatici. Raramente vengono utilizzati per lavori all'aperto, perché il loro impiego al di fuori dell'ambiente organizzato della fabbrica è molto piú complicato.

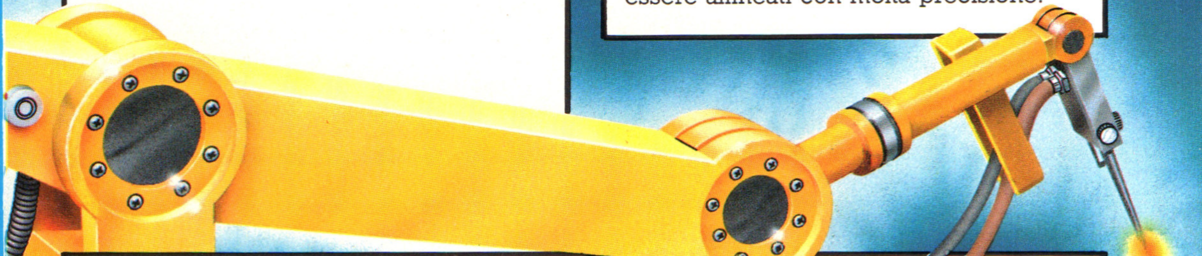
Spesso la fantascenza ci mostra i robot simili agli uomini, invece l'aspetto e le capacità di un robot industriale dipendono dal tipo di lavoro che deve svolgere: i piú assomigliano a ''braccia'' fissate a terra, perché i loro compiti possono essere eseguiti da fermi. Questi robot ''a braccio'' vengono anche chiamati manipolatori, perché tengono qualcosa ''in mano'', come il saldatore della figura sottostante.

I robot a braccio sono comuni soprattutto nelle fabbriche di automobili, ma si trovano anche in molte altre industrie come quelle elettroniche, meccaniche, e tessili. I lavori per i quali sono piú adatti sono ovviamente quelli ripetitivi.



Robot accurati

E' possibile progettare i robot e programmarli perché svolgano complicati lavori di precisione. Questa figura mostra un robot che dispone fili elettrici secondo un intricato disegno per collegare fra loro veicoli elettrici, per esempio carrelli elevatori. A questo scopo, deve innanzi tutto far passare dei perni nei fori di una scheda seguendo un disegno immagazzinato nella memoria del computer. Affinché il funzionamento sia corretto, i perni e i fori devono essere allineati con molta precisione.

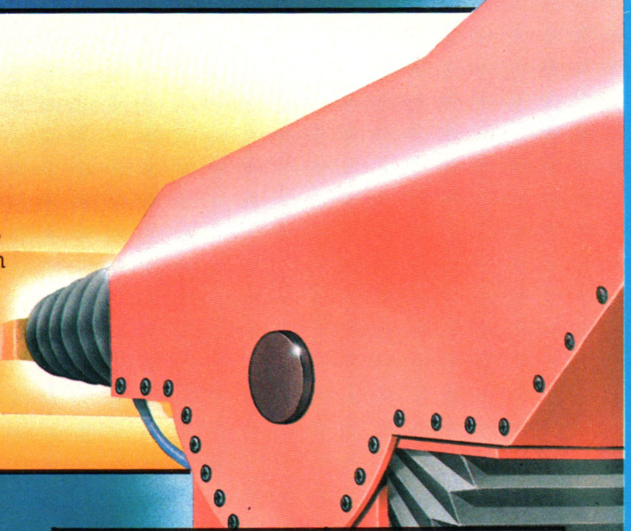


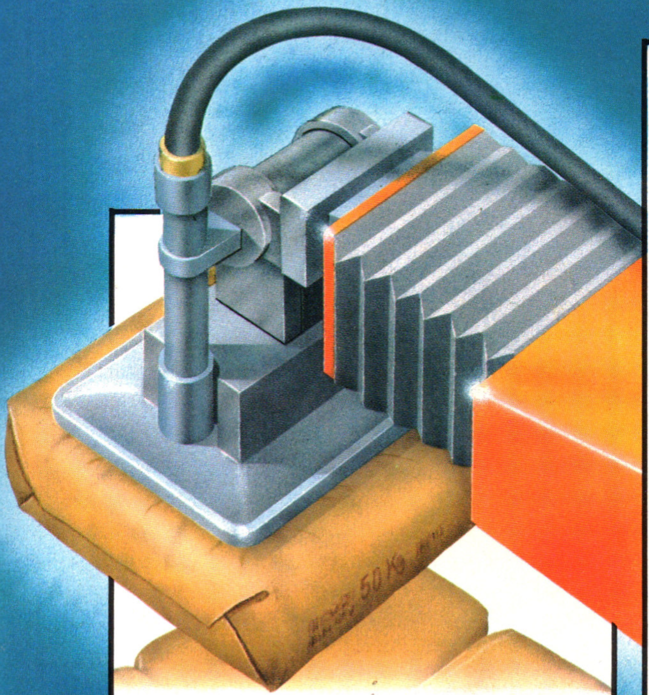
I robot attuali

La maggior parte dei robot esistenti lavora in fabbriche in cui tutto è organizzato attorno a loro. Di solito i robot sono accanto a nastri trasportatori che li ''alimentano'' di lavoro e devono anche essere racchiusi in gabbie protettive perché non provochino danni alle persone. Gli esperti ritengono che fra cinquanta anni sarà possibile fabbricare un robot in grado di lavorare dovunque: si tratterà di un robot molto piú ''intelligente'' di quelli attuali, con una quantità di sensori per poter reagire a una notevole mole di informazioni, o *feedback*, provenienti dall'ambiente circostante. Nemmeno il migliore dei robot oggi esistenti riesce a rispondere con rapidità sufficiente per afferrare, per esempio, una palla. Immagina di doverlo fare tu con un paio di guanti da neve, una mano legata dietro la schiena, gli occhi bendati, i piedi inchiodati a terra e il naso e le orecchie ostruite: la maggior parte dei robot deve basarsi su informazioni anche minori.

Robot resistenti

Molti robot possono svolgere lavori che per le persone sarebbero pericolosi o sgradevoli. I robot, molto resistenti perché metallici, riescono a tollerare condizioni estreme, come un'atmosfera calda e velenosa. Questo robot sta infilando il proprio braccio in una fornace di fonderia. Le sue prestazioni non sono influenzate dal calore, per cui può controllare la produzione di alta qualità, selezionando sempre la temperatura giusta.



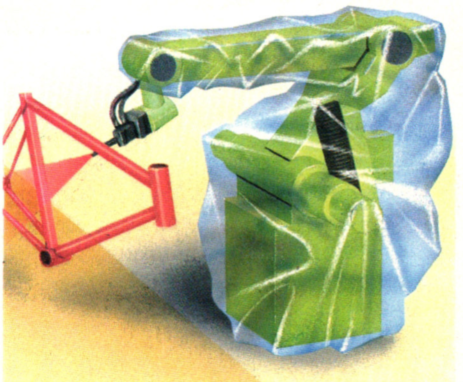


Qual'è la forza di un robot?

La forza di un robot dipende dalla potenza dei suoi motori e dai materiali con cui è costruito: un microrobot domestico fabbricato con un sottile foglio di metallo riesce a sollevare, per esempio, solo il peso di una mela, mentre un grosso robot industriale come quello qui sopra potrebbe alzare oggetti del peso di qualche tonnellata. Un robot del genere potrebbe facilmente sollevare pesi tutto il giorno, mentre anche la piú robusta delle persone finirebbe per stancarsi.

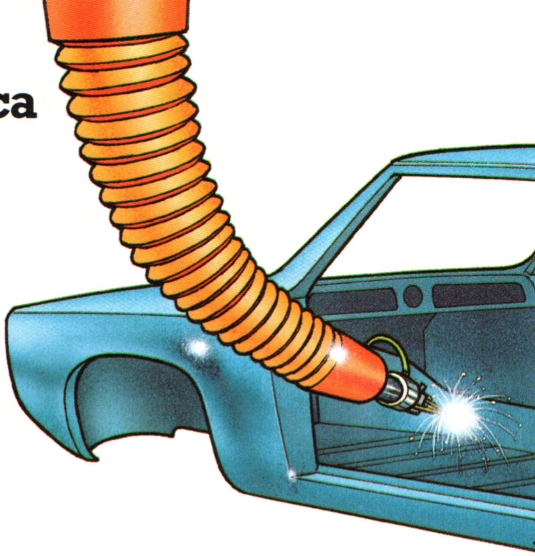
Cosa succede quando qualcosa non funziona?

Molti robot, non avendo sensori, non sono in grado di reagire a situazioni impreviste. Il robot della figura sottostante è controllato da un computer per verniciare a spruzzo telai di bicicletta che passano su un nastro trasportatore; se un telaio cade, il robot continua a verniciare. Un modo per evitare questo inconveniente consiste nell'inserire un interruttore nel nastro che disattivi il robot. Un altro modo per fermarlo è quello di fornirgli sensi elettronici capaci di intuire quanto succede intorno.



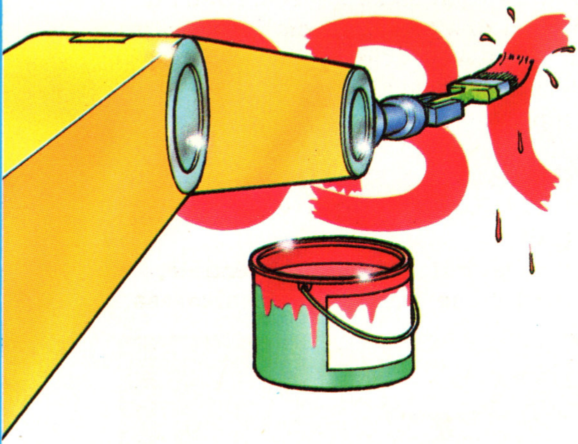
Questo robot è protetto da un rivestimento di plastica perché la vernice non ne ostruisca i meccanismi. Anche altri robot devono essere protetti da rivestimenti speciali.

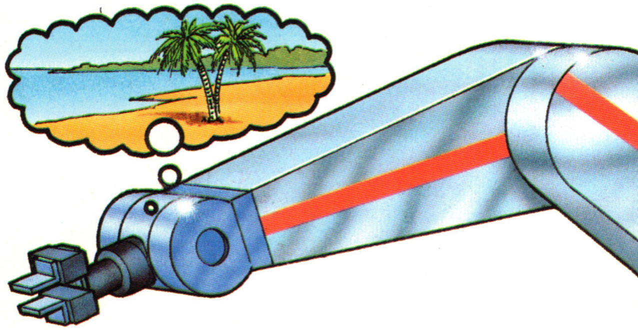
A proposito della robotica



Settant'anni fa la parola ''robot'' era sconosciuta: fu usata per la prima volta da uno scrittore cecoslovacco, Karel Capek (pronuncia *Ciapec*), negli anni venti, in una commedia che trattava di uno scienziato che inventa macchine soprannominato appunto, robot, dalla parola ceca *robota*, che significa ''lavoro da schiavi''. Aveva scelto quel termine perché le macchine dovevano svolgere lavori molto noiosi. Alla fine della commedia, i robot uccidono i padroni umani e assumono il controllo del mondo.

Oggigiorno esistono molti robot, ma sono assai diversi da quelli mostrati nei film e nei libri di fantascienza: invece di uomini metallici superintelligenti, i veri



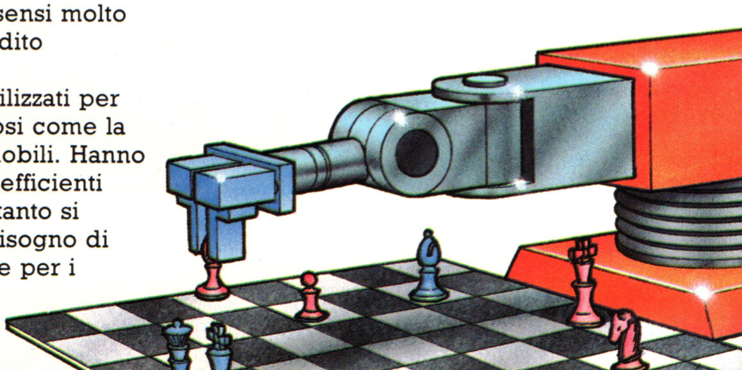


robot sono semplicemente macchine controllate da un computer per svolgere un lavoro prestabilito. Di solito sono sordi, muti, non hanno il senso del gusto, dell'odorato o del tatto né un'intelligenza propria. Però, grazie ai progressi nella tecnologia dei microcircuiti, i robot incominciano a essere provvisti di sensori - (l'''occhio'' di una telecamera o l'''orecchio'' di un microfono), per esempio - che fornisono loro sensi molto limitati, come una vista o un udito elettronici.

I robot vengono spesso utilizzati per lavori pericolosi o molto faticosi come la saldatura dei telai delle automobili. Hanno inoltre il pregio di essere piú efficienti delle persone: sebbene ogni tanto si rompano, i robot non hanno bisogno di vacanze, di dormire o di pause per i pasti.

Essi vengono spesso preferiti ad altri macchinari automatici, perché possono essere programmati per svolgere lavori diversi.

Alcuni robot fanno cose impossibili per gli esseri umani, come lavorare nella zona radioattiva di una centrale nucleare o atterrare su lontani pianeti. Altri, come i microrobot, utilizzati con uno home computer, servono solo per divertimento o per studiare la robotica.



Il sistema operativo

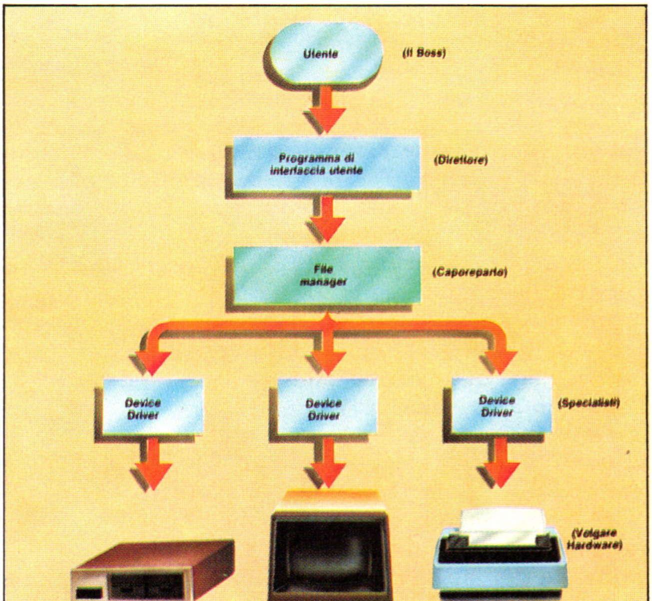
Il calcolatore è essenzialmente una macchina "non intelligente": occorre infatti insegnargli ogni operazione da eseguire. Ecco quindi la necessità del software di base o sistema operativo per gestire le risorse hardware del sistema.

Si tratta di un elemento fondamentale di tutti i moderni computer, per questo motivo è indispensabile capire cosa è, da dove nasce e quale compito è chiamato a svolgere.
Il sistema operativo è un programma che ha il compito di facilitare l'uso delle risorse della macchina (unità centrale, memorie e periferiche) da parte degli utenti e degli altri programmi. Questi non hanno più necessità di conoscere in dettaglio la particolare struttura hardware del calcolatore (ad esempio, come fare per registrare un byte su dischetto): è sufficiente ordinare al sistema operativo di occuparsi del problema.
Il sistema operativo fa parte del software di sistema, cioè quello essenziale per il funzionamento della maccchina. Nei computer più piccoli è residente in ROM, cioè è contenuto in una memoria permanente, che non perde il proprio contenuto in assenza di alimentazione. Si parla in questo caso di firmware: software fornito dalla ditta costruttrice e non modificabile.
In quasi tutti i calcolatori dotati di unità a dischetto (disk drive), il sistema operativo è invece registrato su floppy disk, come qualsiasi altro programma, ed è automaticamente caricato nella memoria del computer (RAM) quando questo viene acceso.

Compiti del sistema operativo

Occorre sottolineare che, per le funzioni svolte, il sistema operativo impiegato è determinante per l'efficienza complessiva di un computer, più di ogni altra caratteristica (hardware compreso). Un S.O., infatti, non si limita alla gestione delle unità a disco, ma ha il compito di ottimizzare, cioè di portare alla massima efficienza, l'uso delle risorse della macchina. Queste comprendono:

- il tempo della CPU (unità centrale);
- la comunicazione con le periferiche (I/O);
- l'impiego della memoria centrale;
- la gestione delle memorie di massa.



I sistemi operativi più diffusi

CP/M - 80

Basato sul processore 8080 e successivamente sullo Z80, è stato il primo sistema operativo di grande successo, dominando il periodo 1978-84. Per esso sono stati scritti innumerevoli programmi, superati in quantità (e qualità) solo dagli attuali per MS-DOS. È ancora oggi usato dagli home computer che utilizzano lo Z80 e siano provvisti di un'unità a disco (es. MSX, AMSTRAD CPC 6128, ecc.)

DOS 3.3

Basato sul microprocessore 6502, non ha avuto l'estensione e il successo sperati, perchè utilizzato esclusivamente dalle macchine APPLE e compatibili.

CP/M - 86

È stato un tentativo di trasportare il CP/M sulle macchine con processore 8086/8088. Implementato rispetto all'originale, ne mantiene comunque una certa rigidità di utilizzo, specie nella gestione dei file, per cui non ha avuto un eccessivo successo.

MS-DOS (PC-DOS)

Basato sul microprocessore 8088 (e 8086), ha seguito (o determinato) il successo delle macchine che utilizzavano tale processore. È attualmente il più diffuso, e per esso sono ormai stati scritti decine di migliaia di programmi, molti dei quali hanno posto le basi dell'attuale elaborazione dati sul PC. Concettualmente derivato dal CP/M, è notevolmente più flessibile sia nella gestione dei file che del video.

UCSD-P

Essendo scritto nel P-Code del Pascal, è stato sviluppato sui più diffusi microprocessori come 6502, 8088, 68000 e altri. Insieme di sistema operativo, ambiente e linguaggio , è costruito in modo rigido, tale da impedire accessi indiscriminati al sistema a alla macchina. Completo e potente, risente di una certa mancanza di documentazione tecnica. Non è attualmente molto utilizzato.

UNIX

È stato scritto in parecchie versioni per i microprocessori più diffusi. La versione più utilizzata è comunque quella per il 68000. Viene utilizzato sia sui PC, sia sui mini che sui grossi computer (mainframe). È relativamente semplice da usare, e permette agli esperti di realizzare compiti molto complessi con la pressione di pochi tasti. Nella versione più potente (XENIX) permette anche la multiprogrammazione. Ritenuto da molti il sistema operativo del futuro, non è però ancora riuscito a sfondare del tutto, anche a causa della concorrenza del MS-DOS.

OS/2

Nato recentemente per meglio sfruttare, rispetto a MS-DOS, le maggiori potenzialità del microprocessore 80286, non ha ancora del tutto convinto il mercato, sia perchè non ancora completamente a punto, sia perchè non ancora dotato di apposito software applicativo che possa allettare i futuri utenti. Sono in molti comunque a prevedere che soppianterà il "vecchio" MS-DOS per imporsi come il sistema operativo standard del futuro.

Vuoi sapere proprio tutto sui migliori videogiochi?

Guida VIDEO GIOCHI

LA GRANDE GUIDA A TUTTI I GIOCHI ELETTRONICI E NON

La prima vera grande guida indipendente a tutti i migliori giochi per computer, console, giochi da bar e altro ancora.
In ogni numero trovi:

- più di 30 giochi al microscopio
- novità e anteprime
- i game da bar più gettonati
- recensioni dei giochi più famosi
- Nintendomania.



GRUPPO EDITORIALE JACKSON

Scegli il meglio: scegli Jackson.

CPC464 e 6128 fantastici computer, fantastici TV!

L. 399.000 + IVA



TUTTO COMPRESO.

CPC464GT 64 Kb RAM con monitor fosfori verdi, tastiera, registratore a cassetta, joystick, 100 programmi/giochi: L. 399.000.+ IVA

CPC464CTM 64 Kb RAM con monitor a colori, tastiera, registratore a cassette, joystick, 100 programmi/giochi: L. 699.000.+ IVA

CPC6128GT 128 Kb RAM con monitor a fosfori verdi, velocissimo disk driver da 3" doppia faccia (180 Kb + 180 Kb), joystick, 50 programmi/giochi: L. 699.000.+ IVA

CPC6128CTM 128 Kb RAM con monitor a colori, velocissimo disk driver da 3" doppia faccia (180 Kb + 180 Kb), joystick, 50 programmi/giochi: L. 899.000.+ IVA

WKS 6128 TV.

Stazione completa composta da: CPC 6128 CTM; Tavolo a ripiani; Sintonizzatore TV; Antenna amplificata.
Tutto a L. 999.000.+ IVA

PRONTO AMSTRAD.

Telefonaci: 02/26410511, avrai ogni informazione; oppure scrivici: Casella Postale 10794 - 20124 Milano.

LI TROVI QUI.

Presso i numerosissimi punti vendita Amstrad. Cerca quello più vicino su "Amstrad Magazine" in edicola, chiedi anche Junior Amstrad la rivita che ti regala i giochi per CPC (troverai molte notizie in più).
Oltre 150 Centri di Assistenza Tecnia.

FANTASTICO, DIVENTA TV COLOR.

Al momento del tuo acquisto puoi trasformare il tuo CPC con monitor a colori in TV color, il tuo TV color, come? Ma è semplice, basta Acquistare il sintonizzatore TV (MP3) a L. 199.000.+ IVA



AMSTRAD

DALLA PARTE DEL CONSUMATORE